

BIG DATA

TP - usage avancé des bases de données et NoSQL

Nicolas LAGAILLARDIE

3 octobre 2018

1 Objectif

Expérimenter le fonctionnement d'Hadoop à travers un exemple : calcul de la fréquence des mots présents dans les ouvrages de l'ABU puis calcule l'index des mots présents dans les ouvrages de l'ABU.

1.1 Définitions

Hadoop Un logiciel open-source pour réaliser des opérations informatiques des manière fiable, évolutif et distribuée.

Pseudo-distributed mode Un mode qui permet d'émuler le fonctionnement d'opérations distribuées sur plusieurs machines sur une même machine. On parle alors de pseudo-cluster.

2 Données expérimentales

ABU Un ensemble de 202 fichiers txt, provenant d'œuvres françaises. Les mots possèdent donc des accents et d'autres caractères, à prendre en considération dans le décompte.

3 Prérequis

L'ensemble des étapes sont détaillées sur le site internet d'Hadoop. Il s'agit d'abord d'avoir les logiciels prérequis : **ssh**, **rsync** et **Java**. Puis il faut assigner à Hadoop le chemin vers **Java**. En effet, Hadoop utilise Java pour fonctionner. Pour ma part, il a été nécessaire que j'utilise **JDK** et non **JSE**, car certaines classes manquaient à l'appel. Nous sommes ensuite prêts à lancer le **pseudo-cluster**.

4 Installation du pseudo-cluster

Comme nous n'avons pas immédiatement accès à un cluster complet, nous allons lancer un pseudo-cluster. C'est-à-dire que nous allons simuler un cluster, avec un *namenode* et un *datanode*.

Nous devons d'abord configurer le port d'utilisation d'Hadoop puis spécifier le nombre de *namenodes* utilisés. Dans notre cas, voici la configuration utilisée :

Modification du fichier *core-site.xml* :

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

Modification du fichier *hdfs-site.xml* :

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

Il faut ensuite se connecter au *ssh* pour que Hadoop puisse communiquer avec ses autres pseudo *datanodes*.

```
$ ssh localhost
```

5 Exécution du *MapReduce*

Nous allons maintenant lancer notre script afin de pouvoir décompter le nombre de mots dans l'ensemble des fichiers txt. Il faut 6 étapes pour y arriver.

- a. Formatage de l'espace de stockage

```
$ bin/hdfs namenode -format
```

- b. Démarrage du *namenode* et du *datanode*

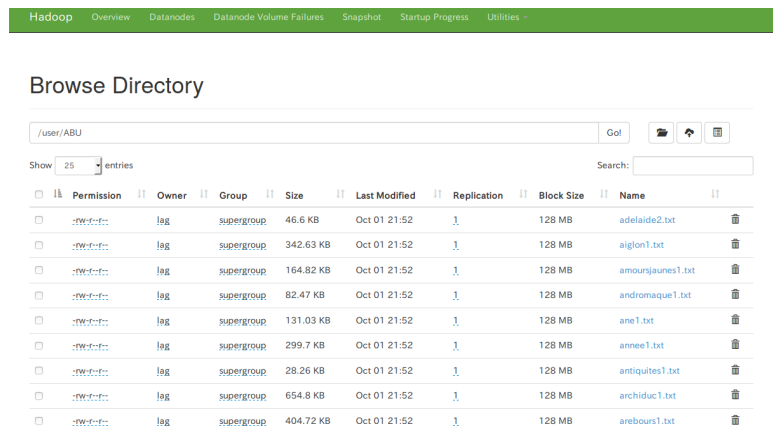
```
$ sbin/start-dfs.sh
```

- c. Création des dossiers *user* sur le cluster

```
$ bin/hdfs dfs -mkdir /user $ bin/hdfs dfs -mkdir /user/Lag
```

- d. Ajout des fichiers dans le cluster

```
$ bin/hdfs dfs -put ABU /user/Lag
```



Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rwxr-xr-x	hqg	supergroup	46.6 KB	Oct 01 21:52	1	128 MB	adelaide2.txt
-rwxr-xr-x	hqg	supergroup	342.63 KB	Oct 01 21:52	1	128 MB	aiglon1.txt
-rwxr-xr-x	hqg	supergroup	164.82 KB	Oct 01 21:52	1	128 MB	amounjaunes1.txt
-rwxr-xr-x	hqg	supergroup	82.47 KB	Oct 01 21:52	1	128 MB	andromaque1.txt
-rwxr-xr-x	hqg	supergroup	131.03 KB	Oct 01 21:52	1	128 MB	ane1.txt
-rwxr-xr-x	hqg	supergroup	299.7 KB	Oct 01 21:52	1	128 MB	annee1.txt
-rwxr-xr-x	hqg	supergroup	28.26 KB	Oct 01 21:52	1	128 MB	antiquites1.txt
-rwxr-xr-x	hqg	supergroup	654.8 KB	Oct 01 21:52	1	128 MB	archiduc1.txt
-rwxr-xr-x	hqg	supergroup	404.72 KB	Oct 01 21:52	1	128 MB	arebours1.txt

FIG. 1 – *Contenu du cluster*

- e. Compiler puis exécuter notre script java

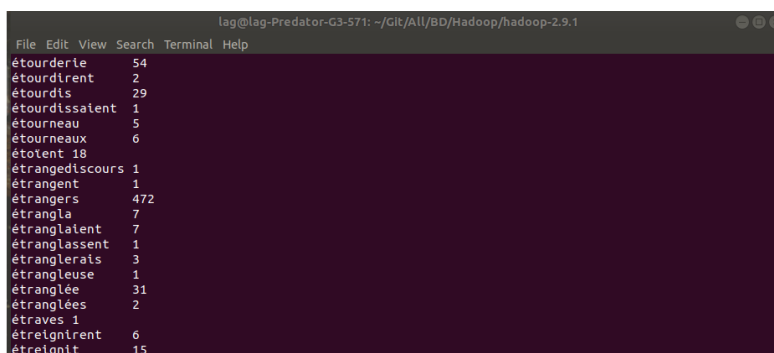
```
$ bin/hadoop com.sun.tools.javac.Main WordCount.java  
$ jar cf WC.jar WordCount*.class  
$ bin/hadoop jar WC.jar WordCount /user/Lag/ABU  
/user/output
```

- f. Récupérer puis afficher les résultats

```
$ bin/hdfs dfs -ls /user/output  
$ bin/hdfs dfs -cat '/user/output/part-*
```


6 Premiers résultats

Suite au lancement de la première version du script, voici les résultats : ils sont globalement satisfaisants (voir Figure 2, mais de nombreux problèmes persistents (voir Figure 3).



File	Edit	View	Search	Terminal	Help
étourderie	54				
étourdirent	2				
étourdis	29				
étourdissaient	1				
étourneau	5				
étourneaux	6				
étotient	18				
étrangediscours	1				
étrangement	1				
étrangers	472				
étrangla	7				
étranglaient	7				
étranglaissent	1				
étranglerais	3				
étrangleuse	1				
étranglée	31				
étranglées	2				
étraves	1				
étraignirent	6				
étraignit	15				

FIG. 2 – *premiers résultats*



3					
dufaure	1				
et	2				
il	1				
pipe	2				
2					
arbre	1				
equateur	1				
aucun	1				
car	1				
eh	1				
et	3				
fil	1				
homme	1				
ils	1				

FIG. 3 – *premières erreurs*

Dans cette deuxième capture d'écran, nous remarquons que certains caractères accentués sont mal représentés. Tandis que d'autres caractères que nous n'avons pas dû prendre en compte dans notre script pour séparer les mots créent des erreurs non voulues.

7 Correction et second résultats

Les modifications possibles concernent la séparation et le comptage des mots des différents fichiers.

Voici une nouvelle séparation possible, qui ne dépend pas de caractères que nous établissons nous-même, mais de l'implémentation même de l'encodage. Ainsi, le code suivant :

```
StringTokenizer itr = new StringTokenizer(
    value.toString().toLowerCase(),
    "“” “n” “r” “f.”;” -’ “” “()!/?[]# =/+ ‘* $ 0123456789
);
while (itr.hasMoreTokens()) {
    word.set(itr.nextToken());
    context.write(word, one);
};
```

Est remplacé par celui-ci :

```
FileSplit fileSplit = (FileSplit)context.getInputSplit();
String filename = fileSplit.getPath().getName();
Path filePath = fileSplit.getPath();
String fileName = filePath.getName();

valueOutFilename = new Text(fileName);

for (String word : StringUtils.split(valueIn.toString())) {
    context.write(new Text(word), valueOutFilename);
}
```

Nous avons ainsi une séparation mieux définie et une concaténation de tous les résultats, au lieu de plusieurs fichiers résultats avec peu de cohérence entre chacun d'eux.

La deuxième partie a consisté à modifier le *Reducer* afin qu'il comptabilise les fichiers contenant les mots et non le nombre d'occurrences de ces derniers. Voici le code que nous avons modifié :

```

private IntWritable result = new IntWritable ();

int sum = 0;

for (IntWritable val: values) {
    sum += val.get();
}

result.set(sum);
context.write(key, result);

```

En ce code :

```

HashSet<String>fileNamesUnique = new HashSet<String>();

for (Text fileName: valuesInFileNames) {
    fileNamesUnique.add(fileName.toString());
}

String fileNamesOut = new String( StringUtils.join(fileNamesUnique,
N° ) );

context.write(keyInWord, new Text(fileNamesOut));

```

Remarquons que j'ai choisi d'utiliser la fonction de hashage présente dans Java, afin d'accélérer le traitement des données. Nous pouvons observer les résultats de l'exécution dans la figure 4 où nous pouvons constater que pour chaque mot, les fichiers le contenant sont affichés à la suite.

```

i.txt fabulistes.txt / chartrai.txt / xibyp2.txt / letfille2.txt / boujgrit.txt / legende2.txt / letblitdivi.txt / comecr2.txt / hmelet.txt / tdm02.txt / reveries3.txt / trnol2.txt / montbardi.txt
/ justice1.txt / medipol.txt / lessolreesi.txt / smarra1.txt / excentlang1.txt / journbloyi.txt / fentini.txt / n702doul2.txt / voyfrani.txt
etres-la chartrai.txt / paspard2.txt
etres-interpre1.txt / chef2.txt / salamb2.txt / monadologie1.txt / especel.txt / lettresjuives231.txt / humlille3.txt / colliergriffesi.txt / supplen2.txt / unevie2.txt / medita1.txt / m
ieromeg3.txt / medit3.txt / lepetitchose1.txt
is roburi1.txt
ile leiphil.txt / educati1.txt / nddp1.txt / bretagne1.txt / contemph2.txt / contrati.txt / ren87922.txt / journalism1.txt / ballades1.txt / mousque1.txt / pomesadvi.txt / balloni.txt / giblasi.txt
/ lepetitchose1.txt / colombi.txt / confessions1.txt / belami2.txt / tdm02.txt / especel.txt / bounty1.txt / reveries3.txt / satani.txt / unevie2.txt / marie1.txt / bouvard2.txt / journbloyi.txt / lep
medeti1.txt / medita1.txt / salamb1.txt / propriete1.txt
ile... Equateur...Fawesee balloni.txt
ile... lepetitchose1.txt
ile: journallami.txt
iles daphni1.txt / candide1.txt / bovary3.txt / vignypoies1.txt / especel.txt / bounty1.txt / balloni.txt / bretagne1.txt
iles... cariboui.txt
llettes ulenspiegel1.txt
lot, tdm022.txt / cariboui.txt // roburi1.txt
ls tomeinalherbel1.txt
ls lettresjuives11.txt
ls duplecor2.txt / contemph2.txt
lots duplecor2.txt / liaisons3.txt / ulenspiegel1.txt
lotant voltgone1.txt / daphni1.txt / colombi.txt / confessions1.txt / chef2.txt / volpofini.txt / reveries1.txt / lettresrecrets1.txt / unevie2.txt / liaisons3.txt / bouvard2.txt / cariboui.txt // jour
nbloyi.txt / ulenspiegel1.txt / raisons1.txt / legende21.txt / pensesse1.txt / balloni.txt / giblasi.txt / voyfrani.txt / voylun3.txt / cyranos1.txt / propriete1.txt / ecole1.txt
lot theorlepy1.txt / pascas1.txt / nddp1.txt / historiettes2.txt / pascaldivi.txt / bretagne1.txt / mousque1.txt / ren05181.txt / vignypoies1.txt / illustoni.txt / giblasi.txt / cinn2.txt / ren0
501.txt / tomeinalherbel1.txt / germinali.txt / echoursi.txt / algoni.txt / colliergriffesi.txt / pensopitti.txt / champli.txt / septfem1.txt / maxime2.txt / lettresrecrets1.txt / germinali.txt / lett
resjuives451.txt / roupei.txt / lettresjuives11.txt / liaisons3.txt / erenovi.txt / preuesi.txt / consider1.txt / anei.txt / voltgome1.txt / lettresjuives231.txt / ren03982.txt / contrati.txt / strgdp1
.txt / mam02.txt / rouspene1.txt / methode1.txt / chartrai.txt / ulenspiegel1.txt / pomesad1.txt / cyranos1.txt / britanico1.txt / confessions1.txt / hmelet.txt / dicollitot.txt / amnei.txt / contemp
lad.txt / dicotpoi.txt / diablom1.txt / reveries3.txt / lessolreesi.txt / homecu1.txt / pucelle1.txt / rayons1.txt / marie1.txt / pensesseXX1.txt / avare2.txt / voyfrani.txt / ruelb01.txt / pascalpettis
1.txt
lot algoni.txt
lot tomeinalherbel1.txt / ulenspiegel1.txt
lot chef2.txt / licausi1.txt / becassi1.txt / letph11.txt / historiettes2.txt / bretagne1.txt / quatrevl1.txt / ren87922.txt / cleves2.txt / mousque1.txt / ren05181.txt / vignypoies1.txt / illustoni1
.txt / leoloi1.txt / giblasi.txt / medit3.txt / lettresjuives0781.txt / ren09041.txt / tomeinalherbel1.txt / chahert3.txt / germinali.txt / algoni.txt / pensopitti.txt / champli.txt / septfem2.txt / maxime
2.txt / satani.txt / lettresrecrets1.txt / lejoul.txt / germinali.txt / diablencor2.txt / lettresjuives451.txt / roupei.txt / lettresjuives11.txt / thoudice1.txt / liaisons3.txt / raisons1.txt / avare2
.txt / propriete1.txt / volpofini.txt / educati1.txt / lettresjuives231.txt / lecdi1.txt / lutrini.txt / ren03982.txt / contrati.txt / gulzeuri.txt / journallami.txt / mam02.txt / methode1.txt / ulenspieg
el1.txt / chartrai.txt / dom3.txt / boujgrit.txt / scapln2.txt / comecr2.txt / colombi.txt / tartuf2.txt / belami2.txt / confessions1.txt / chandre2.txt / reveries3.txt / montbardi.txt / lessolreesi1.t
xt / commerce1.txt / candide1.txt / pucelle1.txt / marie1.txt / fadette1.txt / pensesseXX1.txt / n702doul2.txt / voyfrani.txt / pascalpettisi.txt
loterant liaisons3.txt
loterant ren09041.txt / commerce1.txt / chartrai.txt / neveu2.txt / cleves2.txt / mam02.txt
loterant lettresjuives451.txt / lettresjuives231.txt
loterez satani.txt
lotez-lul chartrai.txt
lotez-moi ulenspiegel1.txt
lotez-vous fabulistes.txt / bretagne1.txt / champli.txt / lettresrecrets1.txt
loter... homecu1.txt
lotit daphni1.txt / lettresjuives451.txt / lettresjuives11.txt / historiettes2.txt / n702doul2.txt
lotmes giblasi.txt / medita1.txt
lot nddp1.txt / monallari.txt / historiettes2.txt / pascaldivi.txt / amoursjaunes1.txt / bretagne1.txt / quatrevl1.txt / opinions3.txt / etutti.txt / cleves2.txt / mousque1.txt / supplen2.txt / gibl
asi.txt / cinn2.txt / medit3.txt / lettresjuives0781.txt / chahert3.txt / tomeinalherbel1.txt / germinali.txt / algoni.txt / colliergriffesi.txt / lettresrecrets1.txt / satani.txt / lettresjuives451.txt
/ roupei.txt / unevie2.txt / liaisons3.txt / lettresjuives11.txt / legende21.txt / salamb1.txt / propriete1.txt / educati1.txt / lecdi1.txt / lettresjuives231.txt / contrati.txt / strgdp1.txt / rouspene
1.txt / methode1.txt / chartrai.txt / ulenspiegel1.txt / dom3.txt / voylun3.txt / lepetitchose1.txt / colombi.txt / daphni1.txt / horla3.txt / confessions1.txt / belami2.txt / legendi.txt / morte2.txt
/ reveries3.txt / trnol2.txt / homecu1.txt / pucelle1.txt / fadette1.txt / pensesseXX1.txt / n702doul2.txt / avare2.txt / pascalpettisi.txt
loter... bovary3.txt / legendi.txt / pensesseXX1.txt / scapln2.txt
loter... confessions1.txt
logging-pradehor-05-0791--GJt/All/NO/hadoop/hadoop-2.9.15 02;c02;c02;c02;c02;c02;d

```

FIG. 4 – *Index des mots*

8 Conclusion

Hadoop est un puissant système et ce TP a pu démontrer une des utilisations possibles. A l'aide d'un *simple* script Java, nous avons été en mesure de créer un index des mots présents dans 202 fichiers txt. Nous avons pu aussi utiliser un compteur de mots performant mais légèrement défaillant, que nous avons corrigé par la suite.

Par ailleurs, l'installation et l'utilisation d'Hadoop est bien documentée et assez simple en mode Pseudo-cluster.